AFRL-SR-AR-TR-05-

0291

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
1 March 2004 - 31 December 2004

4. TITLE AND SUBTITLE
Parallel Simulation of Moving Boundary Flow Using Overset Adaptive Cartesian/Prism Grids and DES

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-04-1-0053

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Z. J. Wang

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Machanical Engineering
Michigan State University
2555 Engineering Building
East Lansing, MI 48824

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
4015 Wilson Blvd
Mail Room 713
Arlington, VA 22203

JWM

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Two major tasks have been undertaken in this year: the development of OverCart - the overset Cartesian/prism grid generator and adaptor, and the enhancement of the flow solver to handle moving overset Cartesian/prism grids. Satisfactory progresses have been made in both tasks, and we are on target and within budget to accomplish all the planned research and development efforts. More specifically, the following activities have been carried out from March 1 to September 30, 2004: 1. An extensive effort has been made in makig the prism grid generation more robust. A comprehensive literature search was carried out to identify the most recent development in this area, and new algorithms are designed and tested. As a result, the prism grid generation algorithm is much more robust, and was capable of tackling vet) complex geometries; 2. The hole cutting and donor cell finding have been implemented in the OverCart software. Octree-based and ADT tree-based algorithms have been employed to speed up the search operations.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU		Z. J. Wang
					19b. TELEPHONE NUMBER *(Include area code)* 517-432-9132

Parallel Simulation of Moving Boundary Flow Using Overset Adaptive Cartesian/Prism Grids and DES

By

Z.J. Wang
Department of Mechanical Engineering
Michigan State University
2555 Engineering Building
East Lansing, MI 48824
Tel: 517-432-9132
Fax: 517-353-1750

September 30, 2004

Annual Performance Report
For the Period of March 1, 2004 – September 30, 2004
Air Force Office of Scientific Research
4015 Wilson Boulevard, Room 713
Arlington, VA 22203-1954

Contract Number: **FA9550-04-1-0053**
Program Director: **Dr. Fariba Fahroo**

RESEARCH OBJECTIVES

The ultimate objective of the effort is to develop and demonstrate a parallel simulation environment that can tackle moving-body problems such as munitions/aircraft separations accurately, efficiently and in a nearly automated fashion. The environment will provide a better understanding of the complex aerodynamics created by moving boundaries.

More specifically, the following objectives are set for this project:

1. Improve the solution accuracy and efficiency through the use of overset adaptive Cartesian/prism grids and solution based grid adaptations.

2. Enhance the flow solver to handle arbitrary overset moving grids including a stationary adaptive Cartesian grid and possibly multiple moving prism grids.

3. Improve the accuracy for highly turbulent unsteady separated flows encountered in moving boundary problems with Detached Eddy Simulation (DES). The DES approach will be evaluated with documented experimental data.

4. Couple the grid generator/grid adaptor/hole cutter with the parallel flow solver using the message passing interface standard, MPI-2, and enable moving boundary problems to be carried out with minimum user interferences.

The objectives remain the same as those outlines in the original proposal. Satisfactory progresses have been made in the past year. We anticipate no technical obstacles.

STATUS OF EFFORT

Two major tasks have been undertaken in this year: the development of OverCart - the overset Cartesian/prism grid generator and adaptor, and the enhancement of the flow solver to handle moving overset Cartesian/prism grids. Satisfactory progresses have been made in both tasks, and we are on target and within budget to accomplish all the planned research and development efforts. More specifically, the following activities have been carried out from March 1 to September 30, 2004:

1. An extensive effort has been makde in making the prism grid generation more robust. A comprehensive literature search was carried out to identify the most recent development in this area, and new algorithms are designed and tested. As a result, the prism grid generation algorithm is much more robust, and was capable of tackling very complex geometries;

2. The hole cutting and donor cell finding have been implemented in the OverCart software. Octree-based and ADT tree-based algorithms have been employed to speed up the search operations. The implementation has been tested and validated with several multi-body cases;

3. The MUSIC flow solver has been enhanced to handle arbitrary overset adaptive Cartesian/prism grids. The MUSIC flow solver reads the donor cell information from the computational grid, and a new boundary condition is implemented to perform solution interpolation. It is made sure that the interpolation is cell-wise linear to match the flow solver accuracy.

MAJOR ACCOMPLISHMENTS

1. Significance of the Present Research

Development of improved weapon systems requires better understanding of the complex aerodynamics created by moving boundaries. Carriage and release of conventional weapons from aircraft, aerial refueling, and formation flying of two or more aircraft fall in this category. The flow problems involving moving boundaries are very challenging to compute because the resultant flow field is intrinsically unsteady. Modeling such flow problems is made even more difficult due to the complex geometries and the highly turbulent nature of the unsteady flows.

In almost all moving boundary flow simulations carried out so far, the Reynolds-Averaged Navier-Stokes (RANS) approach with turbulence models has been used. These models were calibrated according to turbulent boundary layers. The use of the RANS approach for highly separated flows can cause large errors [3-4].

The ultimate objective of the effort is to develop a computational tool that significantly improves on the current state-of-the-art methodology for moving-body problems. We employ an overset adaptive Cartesian/prism grid method for arbitrary moving boundary problems. In addition, a promising hybrid RANS/LES approach named Detached Eddy Simulation (DES) [2] is implemented to improve the solution accuracy for highly separated unsteady flows. Preliminary computations with DES have shown significant improvements in accuracy over RANS [3-4]. Therefore, the main advantages of the proposed methodology for moving-body applications are as follows: (i) a significant reduction in the problem set-up time; (ii) a significant reduction in the calculation turn-around time; (iii) accuracy enhancements due to the use of adaptive grid refinement and DES.

2. Robustness Improvement on Prism Grid Generation for Complex Geometries

The generation of prism grids around complex geometries is notoriously difficult because of robustness problems. For convex body shapes, it is trivial to generate body-fitted prism grids because one can easily "extrude" the surface grid in the normal direction. However, for non-convex or concave bodies, grid lines can cross each other, resulting in invalid computational grid cells. Although a "fool-proof" algorithm for prism grid generation is desired, it remains a goal to be achieved probably for many years to come, if ever. Nevertheless, a significant effort has been made in this project to improve the robustness of the present prism grid generation algorithm.

The input to prism grid generation is a triangular surface grid. Then this triangular grid is marched away from the surface, thus resulting in structured prism layers in the marching direction. The accuracy of Navier Solver solver depends to a great extent on the quality of the grids produced. If strict orthogonality is imposed while generating prism grids, the marching vector is the outer normal to the plane at that point. In practice, the marching vector is offset from the outer normal by a small angle. This is done to avoid intersections in concave regions. The steps taken in the prism grid algorithm are described next.

2.1 Generation of Marching Vectors

This is the quintessential aspect of prism grid generation. As discussed earlier, orthogonality of the grids and non intersecting marching vectors are contradicting design philosophies. So a balance needs to be struck between the above contradicting conditions.

Let us now define a new term called 'manifold'. The group of triangular faces sharing a common node is termed a manifold. Common sense tells that the marching vector at a particular node should not make an angle more than 90 degrees with any of the face normals of the manifold. If the above criterion is violated, the tip of the marching vector is not visible from all the points in the manifold. This results in intersections.

Going by the above logic, it can be said that the maximum of the angles between the marching vector at a node and the face normals of the manifold needs to be as small as possible. In other words, the marching vector needs to be chosen in a manner that minimizes the maximum angle between the marching vector and the face normals of the manifold. The procedure for obtaining the marching vector at a node is as follows. A good initial guess for the marching vector is to use a weighted average of the face normals of the manifold. These marching vectors obtained are locally further refined to reduce their maximum angle.

In order to prevent the intersection of marching vectors at concave surfaces, the marching vectors obtained above are smoothed with a Laplacian smoothing. This smoothing function is invoked many times to obtain a smooth contour.

2.2 Computation of the Layer Thickness

Once the marching vectors are determined, the nodes need be positioned at the new layer. One of the many traits of a good grid generating algorithm is to reduce the curvature of the front from one layer to the next layer. Thus it would be unwise to have a constant layer thickness at all nodes in a particular layer. Concave nodes need to be marched faster and convex nodes need to be marched slower. However, care must be taken to ensure that the thickness ratio between two adjacent nodes should not be less than 0.5. The above needs to be implemented to obtain an accurate solution.

However, it is necessary for the average thickness (the value obtained by averaging the thickness over all the nodes) to increase exponentially with increasing layers. In other words, if the average thickness at the first layer is Δn_1, and the average thickness at the *ith* layer is Δn_i then

$$\Delta n_i = \Delta n_1 s^{i-1}$$

Where s is the stretching factor and is generally less than 1.2.

An empirical relation for the layer thickness at node j of layer i Δn_{ij} was formulated as

$$\Delta n_{ij} = \Delta n_i (r + (1-r)\cos(\alpha_j))$$

Where α_j is the maximum angle between the marching vector at node j and the normals of its manifold, r is a parameter which is less than one (generally 0.5) for convex nodes and greater than one (generally 1.87) for concave nodes. Thus the maximum value for the layer thickness is

1.87 times the average thickness and the minimum value for the layer thickness is 0.5 times the average thickness.

2.3 Intersections Checking

In most cases, the efficiency and accuracy of the flow solver depends on the quality of the grid generated. Even extremely robust solvers will fail to produce acceptable results if the grid generated is invalid. A grid is said to be invalid if

- There are intersections;
- There are negative areas/volumes.

Actually negative areas and negative volumes are generally created due to intersections. So the main issue here is to identify intersections. The following diagram shows a prism with no intersections. Triangle ABC is the triangular face at the current layer. Triangle DEF is the triangular face at the next layer

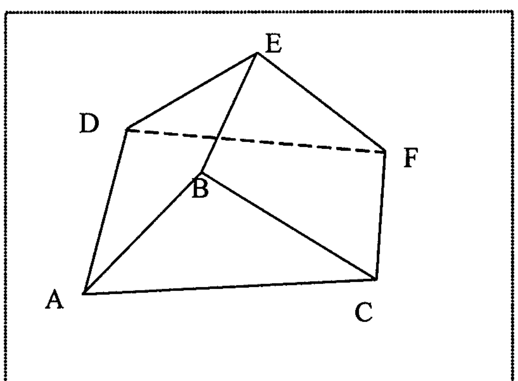


Figure 1. A Prism with No Intersections

Intersections can occur when either of the below occurs

- Node A and Node D do not lie on the same side of the planes BCEB and BCFB
- Node B and Node E do not lie on the same side of the planes ACDA and ACFA
- Node C and Node F do not lie on the same side of the planes ABDA and ABEA

2.4 Actual Working of the Algorithm

After the marching vectors are determined, the curvature of the surface at each node is computed. Then the marching vectors are smoothed. After that, the thickness of the layer at each node is determined. Corrective measures are taken if the ratio of the thickness between two adjacent nodes falls below 0.5 or above 2. Finally the position of the nodes in the next layer is obtained.

The algorithm discussed till now is still not perfect. As per the algorithm, the nodes in the concave region get closer with advancing layers. This results in the triangles becoming increasingly skewed with advancing layers and hence causing intersections between the marching vectors. In order to circumvent the above, smoothing of the nodes in the new layer is also done. At first, a Laplacian smoothing step was attempted. This smoothing did not work out very well especially at concave regions. After some literature survey, a new type of smoothing (called mean filtering) which works well in the concave regime was found and implemented.

2.5 Demonstration Examples

The algorithm developed above was used to generate prism grids for many real life geometries like store (missile with fine), missile, wing-body configuration and a complete aircraft. The prism grid generated for a store is shown in Figure 2. The sharp concave corners are handled by the new algorithm satisfactorily.

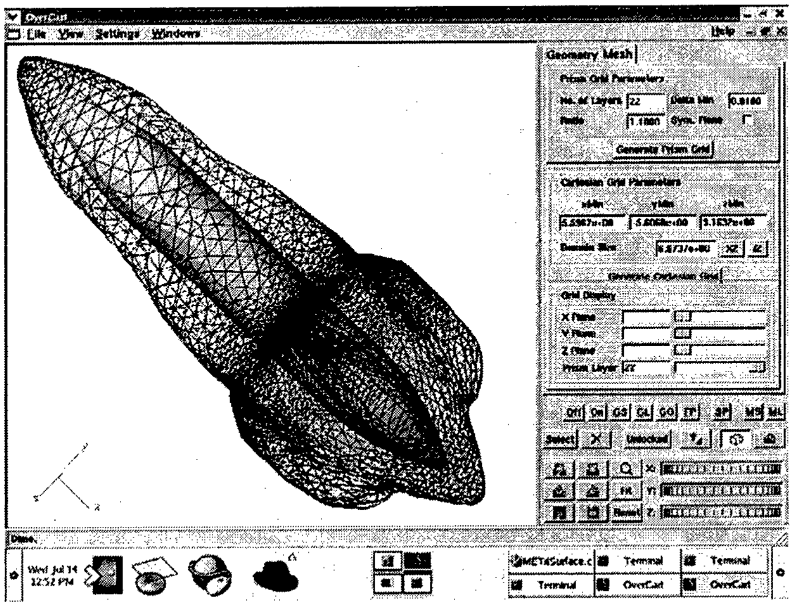


Figure 2. Prism Grid for the Store

3. Automated Hole Cutting, and Donor Cell Identification

The use of overset adaptive Cartesian and prismatic grids has the potential of handling arbitrary moving boundary problems without user interference. A critical element in achieving this level of automation is an automated hole-cutting algorithm, in which invalid Cartesian grid cells (cells inside a solid body) are excluded from the calculation, and donor cells are identified for hole boundary cells and outer boundary cells (the outer most layer in the prismatic grid). A schematic of the hole-cutting operation with definition of terminology is shown in Figure 3.

An automated hole cutting algorithm [5-6] has been implemented in OverCart. The efficiency of the hole-cutting algorithm is critical since many steps of the hole-cutting operation will be performed in a moving boundary problem as the prismatic grids move in the flow field. To achieve the maximum efficiency for the hole-cutting algorithm, two search trees have been used extensively. One is the *Octree* used in generating the adaptive Cartesian grid, and the other is an alternating digital tree (ADT) [1] for the bounding boxes of the prismatic cells. The use of the *Octree* to speed up search operations is another significant advantage of using the adaptive Cartesian grid for moving boundary flows. The hole-cutting algorithm consists of the following steps:

1 Outer boundaries of the prismatic grids are used to generate hole-boundaries in the adaptive Cartesian grids. Again the *Octree* is used to identify Cartesian cells which intersect the outer boundaries;

2 Generate a list of hole boundary cells in the adaptive Cartesian grid, and blank all the Cartesian cells which are inside the outer boundary. Use the ADT tree to find the prismatic cells, which bound the centroids of hole boundary cells. The prismatic cells bounding the centroids of the hole boundary cells are also called donor cells.

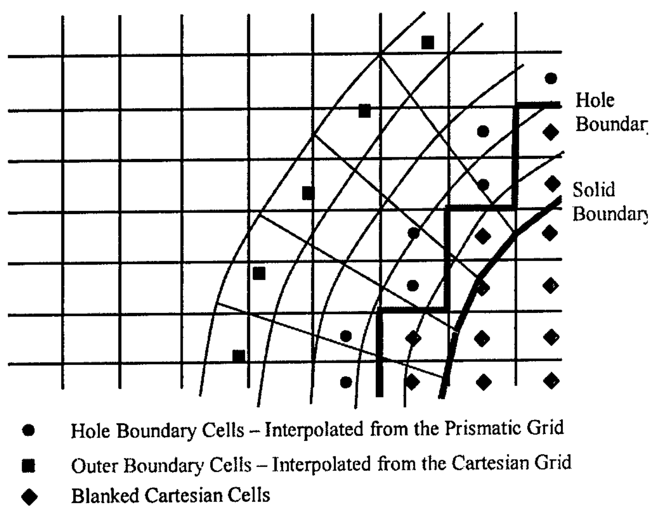
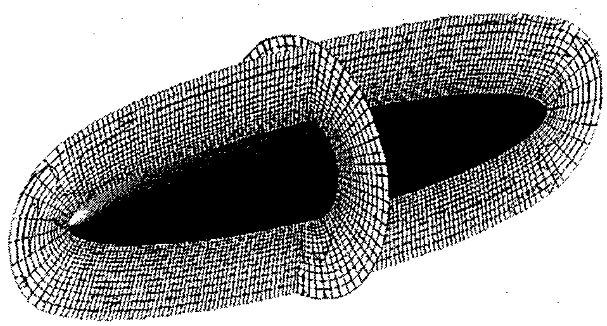


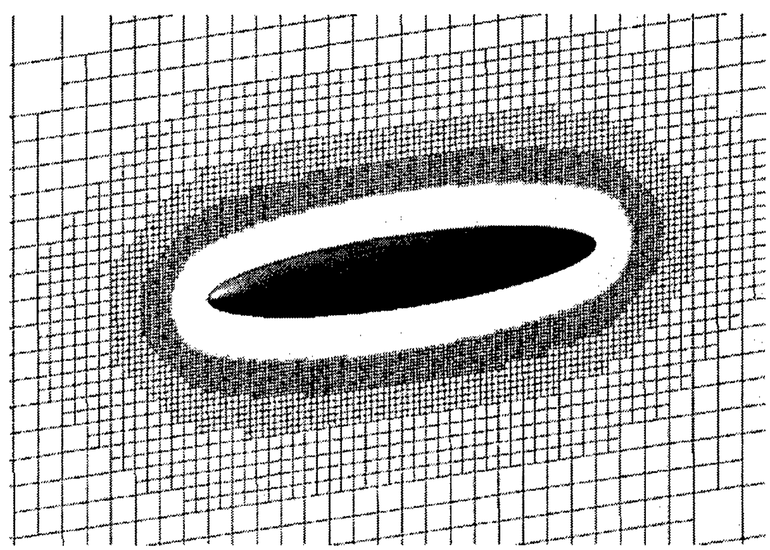
Figure 3. Schematic of Hole-Cutting

3 Generate a list of outer boundary cells, and use the *Octree* (again!) to identify the Cartesian cells which bound the cell-centroids of the outer boundary cells of the prismatic grids. These Cartesian cells are also called donor cells;

An example of hole-cutting is shown in Figure 4, which displays an overset Cartesian/prism grid for a prolate shape.



(a) Prism Grid



(b) Chimera Hole Generated in the Cartesian Grid

Figure 4. Illustration of Hole-Cutting in a Overset Cartesian/Prism Grid

4. Enhancement of Flow Solver to Handle Overset Grids

The MUSIC flow solver [9] has been enhanced to handle arbitrary overset grids. First, a list of cells in each domain, which require data interpolation from other domains are read in from the computational grid generated by OverCart. For each interpolation cell, the donor cell information, including the domain of the donor cell and the cell index, is also read in. Then in the flow solver, flow solutions at these interpolation cells are obtained from the donor cells assuming the solutions are cell-wise linear in the donor cells. This interpolation procedure is fully compatible with the data reconstruction used in the flow solver, and is second-order accurate.

Several sample computations have been performed to ensure that the solution interpolation is carried out properly. Shown in Figure 5 is the computed pressure contours with the grid shown in Figure 4. It appears the data interpolation algorithm has been implemented properly.

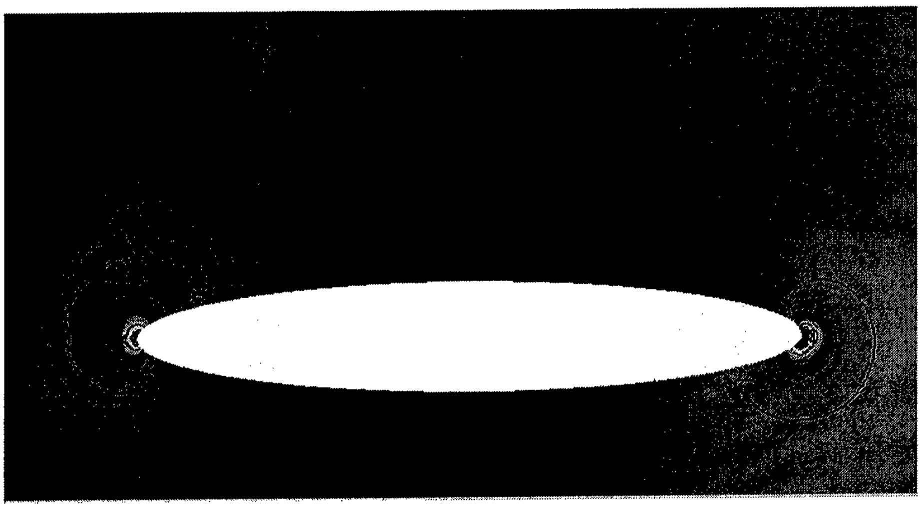


Figure 5. Pressure Contours Computed with an Overset Cartesian/Prism Grid

PERSONNEL SUPPORTED

The project supported one summer month of the PI, Dr. Z.J. Wang, and partially supported two graduate students, Ravi Kannan (Master) and Yuzhi Sun (Ph.D.).

PUBLICATIONS

Two papers have been published or accepted. Other publications are planned.

1. Z.J. Wang and R. Kannan, "An Overset Adaptive Cartesian/Prism Grid Method for Moving Boundary Flow Problems," accepted by the 43th AIAA Aerospace Sciences Meeting and ExhibitJan. 2005.
2. L. Zhang and Z.J. Wang, "A Block LU-SGS Implicit Dual Time-Stepping Algorithm for Hybrid Dynamic Meshes," *Computer & Fluids* Vol. 33, pp. 891-916, 2004.

INTERACTIONS

The PI gave the following talks in several scientific meetings:

1. "Extension of the Spectral Volume Method to the 2D Navier-Stokes Equations," ICOSAHOM 2004, Brown University, June 21-25 2004.
2. "Direct Simulation of Surface Roughness Effects with RANS and DES Approaches on Viscous Adaptive Cartesian Grids," 34th AIAA Fluids Dynamics Conference, June 28 – 1 July 2004, Portland, Oregon.
3. "Computation of Aeroacoustic Waves with High Order Spectral Volume Method," 3th International Conference on CFD, Toronto, Canada, July 12-16 2004.

PATENTS

None.

REFERENCES

1. Bonet, J. and Peraire, J., "An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Search and Intersection Problems," International Journal for Numerical Methods in Engineering, Vol. 31, pp. 1-17, 1991.
2. Spalart, P.R., Jou, W.H., Strelets, M. and Allmaras, S.R., "Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach," First AFOSR International Conference on DNS/LES, Ruston, LA, 1997.
3. Forsythe, J.R., Squires K.D., Wurtzler, K.E. and Spalart, P.R., "Detached-Eddy Simulation of Fighter Aircraft at High Alpha," AIAA Paper No. 2002-0591.
4. Tomaro, R. F., Witzeman, F. C., Strang, W. Z., "Simulation of Store Separation for the F/A-18C Using $Cobalt_{60}$," Journal of Aircraft, Vol. 37, No. 3, May-June 2000, pp. 361-67.
5. Wang, Z.J. and V. Parthasarathy, "A Fully Automated Chimera Methodology for Multiple Moving Body Problems," *International Journal for Numerical Methods in Fluids*, Vol. 33, No. 7, pp 919-938, 2000.
6. Wang, Z.J. and Bayyuk, S.A., "An Automated Adaptive, Unstructured, Cartesian-Prism-Based Technique for Moving-Boundary Simulations," in Proceedings of 6th International Conference on Numerical Grid Generation in Computational Field Simulations, London, July 1998.
7. Wang, Z.J. and Chen, R.F., "Parallel Adaptive Cartesian Grid Method for Viscous Flow with Complex Geometries," *CFD Journal*, Vol. 10, No. 4, pp.575-584, 2002.